
RIL_GetDataCompression

HRESULT RIL_GetDataCompression(HRIL *hRil*)

Retrieves data compression options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILDATACOMPINFO** structure.

RIL_GetDevCaps

HRESULT RIL_GetDevCaps(HRIL *hRil*, **DWORD** *dwCapsType*)

Retrieves specified device capabilities

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwCapsType

type of caps class to retrieve

Comments

Asynchronous.

dwCapsType (**RIL_CAPSTYPE_**) *lpData*

* **_DIAL** points to an **RILCAPSDIAL** structure

* **_DTMFDURATIONRANGE** points to an **RILRANGE** structure (values in milliseconds)

* **_CALLMGTCMDS** points to **DWORD** containing a combination of **RIL_CAPS_CALLCMD_** constants

* **_BEARERSERVICE** points to an **RILCAPSBEARERSVC** structure

* **_RLP** points to an array of **RILAPSRLP** structures

* **_EQUIPMENTSTATES** points to **DWORD** containing a combination of **RIL_CAPS_EQSTATE_** constants

* **_PBSTORELOCATIONS** points to **DWORD** containing a combination of **RIL_CAPS_PBLOC_** constants

* **_PBINDEXRANGE** points to an **RILRANGE** structure

* **_PBENTRYTEXTLENGTH** points to a **DWORD**

* **_MSGSERVICETYPES** points to **DWORD** containing a combination of **RIL_CAPS_MSGSVCTYPE_** constants

* **_MSGMEMORYLOCATIONS** points to an **RILCAPSMGSMEMORYLOCATIONS** structure

* **_BROADCASTMSGLANGS** points to **DWORD** containing a combination of **RIL_CAPS_DCSLANG_** constants

* **_MSGCONFIGINDEXRANGE** points to an **RILRANGE** structure

* **_MSGSTATUSVALUES** points to **DWORD** containing a combination of **RIL_CAPS_MSGSTATUS_** constants

* **_PREFOPINDEXRANGE** points to an **RILRANGE** structure

* **_LOCKFACILITIES** points to **DWORD** containing a combination of **RIL_CAPS_LOCKFACILITY_** constants

* ***_LOCKINGPWDLENGTHS*** points to an array of **RILCAPSLOCKINGPWDLENGTH** structures
 * ***_BARRTYPES*** points to **DWORD** containing a combination of **RIL_CAPS_BARRTYPE_** constants
 * ***_BARRINGPWDLENGTHS*** points to an array of **RILCAPSBARRINGPWDLENGTH** structures
 * ***_FORWARDINGREASONS*** points to **DWORD** containing a combination of **RIL_CAPS_FWDREASON_** constants
 * ***_SIMTOOLKITNOTIFICATIONS*** points to a **TBD** SIMTOOLKIT structure
 * ***_INFOCLASSES*** points to **DWORD** containing a combination of **RIL_CAPS_INFOCLASS_** constants
 * ***_HSCSD*** points to an **RILCAPSHSCSD** structure
 * ***_GPRS*** points to an **RILCAPSGPRS** structure

RIL_GetDialedIdSettings

HRESULT RIL_GetDialedIdSettings(HRIL hRil)
 Retrieves the current DialedID settings

Parameters *hRil*
 handle to RIL instance returned by **RIL_Initialize**

Comments Asynchronous. *lpData* points to an **RILDIALEDIDSETTINGS** structure.

RIL_GetEquipmentInfo

HRESULT RIL_GetEquipmentInfo(HRIL hRil)
 Retrieves manufacturer equipment information

Parameters *hRil*
 handle to RIL instance returned by **RIL_Initialize**

Comments Asynchronous. *lpData* points to an **RILEQUIPMENTINFO** structure.

RIL_GetEquipmentState

HRESULT RIL_GetEquipmentState(HRIL hRil)
 Retrieves currently set equipment state

Parameters *hRil*
 handle to RIL instance returned by **RIL_Initialize**

Comments Asynchronous. *lpData* points to an **RILEQUIPMENTSTATE** structure.

RIL_GetErrorCorrection

HRESULT RIL_GetErrorCorrection(HRIL hRil)

Retrieves error correction options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILERRORCORRECTIONINFO** structure.

RIL_GetGPRSAddress

HRESULT RIL_GetGPRSAddress(HRIL hRil, DWORD dwContextID)

Gets the PDP address for a particular context

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwContextID

identifies the context

Comments

Asynchronous. *lpData* points to an array of **WCHAR** values indicating the address.

RIL_GetGPRSAttached

HRESULT RIL_GetGPRSAttached(HRIL hRil)

Gets the GPRS attach state

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **BOOL** indicating attach state.

RIL_GetGPRSClass

HRESULT RIL_GetGPRSClass(HRIL hRil)

Retrieves the current GPRS class

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **RIL_GPRSCCLASS_** constant.

RIL_GetGPRSContextActivatedList

HRESULT RIL_GetGPRSContextActivatedList(HRIL hRil)

Gets the GPRS activation state for all contexts

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
Comments	Asynchronous. <i>lpData</i> points to a RILGPRSCONTEXTACTIVATED indicating activation state.

RIL_GetGPRSContextList

HRESULT RIL_GetGPRSContextList(HRIL hRil)
Retrieves a list GPRS contexts

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
Comments	Asynchronous. <i>lpData</i> points to a RILGPRSCONTEXT structure.

RIL_GetGPRSRegistrationStatus

HRESULT RIL_GetGPRSRegistrationStatus(HRIL hRil)
Retrieves the current GPRS registration status

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
Comments	Asynchronous. <i>lpData</i> points to a RIL_REGSTAT_ constant.

RIL_GetHideConnectedIdSettings

HRESULT RIL_GetHideConnectedIdSettings(HRIL hRil)
Retrieves the current HideConnectedID settings

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
Comments	Asynchronous. <i>lpData</i> points to an RILHIDECONNECTEDIDSETTINGS structure.

RIL_GetHideIdSettings

HRESULT RIL_GetHideIdSettings(HRIL hRil)
Retrieves the current HideID settings

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
Comments	Asynchronous. <i>lpData</i> points to an RILHIDEIDSETTINGS structure.

RIL_GetHSCSDCallSettings

HRESULT RIL_GetHSCSDCallSettings(HRIL hRil)

Retrieves High Speed Circuit Switched Data options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILCALLHSCSDINFO** structure.

RIL_GetHSCSDOptions

HRESULT RIL_GetHSCSDOptions(HRIL hRil)

Retrieves High Speed Circuit Switched Data options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILHSCSDINFO** structure.

RIL_GetLineStatus

HRESULT RIL_GetLineStatus(HRIL hRil)

Retrieves the phone line status

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to **DWORD** containing **RIL_LINESTAT_** constant.

RIL_GetLockingStatus

HRESULT RIL_GetLockingStatus(HRIL hRil, DWORD dwFacility, LPCSTR lpszPassword)

Retrieves locking status for the specified facility

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwFacility

facility to retrieve locking status for (**RIL_LOCKFACILITY_** constant)

lpszPassword

password to retrieve locking status (can be **NULL** if password isn't required; no longer than **MAXLENGTH_PASSWORD** chars)

Comments

Asynchronous. *lpData* points to a **DWORD** containing a **RIL_LOCKINGSTATUS_** constant.

RIL_GetMinimumQualityOfServiceList

HRESULT RIL_GetMinimumQualityOfServiceList(HRIL hRil)

Gets the minimum quality of service profile for all contexts

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **RILGPRSQOSPROFILE** structure.

RIL_GetMOSMSService

HRESULT RIL_GetMOSMSService(HRIL hRil)

Retrieves the preferred SMS service option for mobile originated messages

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **RIL_MOSMSSERVICE_** constant.

RIL_GetMsgConfig

HRESULT RIL_GetMsgConfig(HRIL hRil)

Gets currently set messaging configuration

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILMSGCONFIG** structure.

RIL_GetMsgServiceOptions

HRESULT RIL_GetMsgServiceOptions(HRIL hRil)

Gets currently set messaging service options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILMSGSERVICEINFO** structure.

RIL_GetOperatorList

HRESULT RIL_GetOperatorList(HRIL hRil)

Retrieves the list of available operators

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments Asynchronous. *lpData* points to an array of **RILOPERATORINFO** structures.

RIL_GetPhonebookOptions

HRESULT RIL_GetPhonebookOptions(HRIL hRil)

Retrieves currently set phonebook options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILPHONEBOOKINFO** structure.

RIL_GetPhoneLockedState

HRESULT RIL_GetPhoneLockedState(HRIL hRil)

Retrieves current locked state of the phone

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **DWORD** containing a **RIL_LOCKEDSTATE_** constant

RIL_GetPreferredOperatorList

HRESULT RIL_GetPreferredOperatorList(HRIL hRil, DWORD dwFormat)

Retrieves the list of preferred operators

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwFormat

format to use for the operator names in the list

Comments

Asynchronous. *lpData* points to an array of **RILOPERATORINFO** structures.

RIL_GetRadioPresence

HRESULT RIL_GetRadioPresence(HRIL hRIL, DWORD* dwRadioPresence)

Proxy API to determine if the Radio is present or Not (Is the RIL driver Loaded?)

Parameters

hRIL

handle to RIL instance returned by **RIL_Initialize**

dwRadioPresence

pointer to a **DWORD** (ouput param contains values from **RIL_RADIOPRESENCE_***)

Comments

Synchronous

RIL_GetRegistrationStatus

HRESULT RIL_GetRegistrationStatus(HRIL hRil)

Retrieves the current phone registration status

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **RIL_REGSTAT_** constant.

RIL_GetRequestedQualityOfServiceList

HRESULT RIL_GetRequestedQualityOfServiceList(HRIL hRil)

Gets the requested quality of service profile for all contexts

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **RILGPRSQOSPROFILE** structure.

RIL_GetRLPOptions

HRESULT RIL_GetRLPOptions(HRIL hRil)

Retrieves currently set Radio Link Protocol options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an **RILRLPINFO** structure.

RIL_GetSerialPortHandle

HRESULT RIL_GetSerialPortHandle(HRIL hRil, HANDLE* lphSerial)

Retrieves a serial port handle to be used for data communications

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lphSerial

pointer to the serial port handle

Comments

Synchronous. Client is responsible for closing the handle returned in *lphSerial*.

RIL_GetSerialPortStatistics

HRESULT RIL_GetSerialPortStatistics(HRIL hRil, RILSERIALPORTSTATS* lpSerialPortStats)

Retrieves a serial port handle statistics

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

lpSerialPortStats
pointer to the statistics structure

Comments Synchronous

RIL_GetSignalQuality

HRESULT RIL_GetSignalQuality(HRIL hRil)
Retrieves information about the received signal quality

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

Comments Asynchronous. *lpData* points to a **RILSIGNALQUALITY** structure.

RIL_GetSimRecordStatus

HRESULT RIL_GetSimRecordStatus(HRIL hRil, DWORD dwFileID)
Retrieves SIM Record Status

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

dwFileID
address of the file to read

Comments Asynchronous. *lpData* points to **RILSIMRECORDSTATUS**

RIL_GetSimToolkitProfile

HRESULT RIL_GetSimToolkitProfile(HRIL hRil)
Retrieves SIM Toolkit terminal profile

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

Comments Asynchronous. *lpData* points to an array of **BYTE**s.

RIL_GetSubscriberNumbers

HRESULT RIL_GetSubscriberNumbers(HRIL hRil)
Restrieves information about subscriber numbers

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

Asynchronous. *lpData* points to an array of **RILSUBSCRIBERINFO** structures.

RIL GetSystemTime

HRESULT RIL_GetSystemTime(HRIL *hRil*)

Retrieves the systemtime from the network

Parameters

 $hRil$

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to a **SYSTEMTIME** structure. This feature is currently not used in Stinger and is untested.

RIL GetUserIdentity

HRESULT RIL_GetUserIdentity(HRIL *hRil*)

Retrieves International Mobile Subscriber Identity of the phone user

Parameters

 $hRil$

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* points to an array of **chars**

RIL GPRSAnswer

$$\mathbf{HRESULT} \text{ RIL_GPRSAnswer}(\mathbf{HRIL} \ hRil)$$

Answers an incoming GPRS activation request

Parameters

 $hRil$

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* is *NULL*.

RIL Hangup

HRESULT RIL_Hangup(HRIL *hRil*)

Hangs up all calls currently in progress

Parameters

 $hRil$

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* is *NULL*.

RIL Initialize

HRESULT RIL_Initialize(DWORD dwIndex, RILRESULTCALLBACK pfnResult, RILNOTIFYCALLBACK pfnNotify, DWORD dwNotificationClasses, DWORD dwParam, HRIL* lphRil)

Initializes RIL for use by this client

Parameters

dwIndex

index of the RIL port to use (e.g., 1 for RIL1:)

pfnResult

function result callback

pfnNotify

notification callback

dwNotificationClasses

classes of notifications to be enabled for this client

dwParam

custom parameter passed to result and notification callbacks

lphRil

returned handle to RIL instance

Comments

Synchronous RIL only supports single threaded RIL handles. The RIL validates the application's RIL handle before using it. No application can use/close a RIL handle that it does not own.

RIL InitializeEmergency

HRESULT RIL_InitializeEmergency(DWORD dwIndex, RILRESULTCALLBACK pfnResult, RILNOTIFYCALLBACK pfnNotify, DWORD dwNotificationClasses, DWORD dwParam, HRIL* lphRil)

Initializes RIL for use by this emergency call module

Parameters

dwIndex

index of the RIL port to use (e.g., 1 for RIL1:)

pfnResult

function result callback

pfnNotify

notification callback

dwNotificationClasses

classes of notifications to be enabled for this client

dwParam

custom parameter passed to result and notification callbacks

lphRil

returned handle to RIL instance

Comments

Synchronous

RIL_ManageCalls

HRESULT RIL_ManageCalls(**HRIL** *hRil*, **DWORD** *dwCommand*, **DWORD** *dwID*)

Modifies the state of active, held, and waiting calls

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwCommand

call modification command to be performed (**RIL_CALLCMD_** constant)

dwID

ID of the call to be modified (only for **RIL_CALLCMD_RELEASECALL** and **RIL_CALLCMD_HOLDALLBUTONE**)

Comments

Asynchronous. *lpData* is **NULL**.

RIL_ReadMsg

HRESULT RIL_ReadMsg(**HRIL** *hRil*, **DWORD** *dwIndex*)

Reads a message from the current storage location

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwIndex

index of the message to be read

Comments

Asynchronous. *lpData* points to an **RILMESSAGEINFO** structure.

RIL_ReadPhonebookEntries

HRESULT RIL_ReadPhonebookEntries(**HRIL** *hRil*, **DWORD** *dwStartIndex*, **DWORD** *dwEndIndex*)

Reads phonebook entries from the specified range of indices of the current storage location

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwStartIndex

starting index of the range

dwEndIndex

ending index of the range

Comments

Asynchronous. *lpData* points to an array of **RILPHONEBOOKENTRY** structures.

RIL_RegisterOnNetwork

HRESULT RIL_RegisterOnNetwork(HRIL hRil, DWORD dwMode, const RIOPERATORNAMES* lpOperatorNames)

Registers the ME with a network operator

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwMode

operator selection mode (**RIL_OPSELMODE_** constant)

lpOperatorNames

operator to be selected (can be *NULL* if *dwMode* is **RIL_OPSELMODE_AUTOMATIC**)

Comments

Asynchronous. *lpData* is *NULL*.

RIL_RemoveCallForwarding

HRESULT RIL_RemoveCallForwarding(HRIL hRil, DWORD dwReason, DWORD dwInfoClasses, DWORD dwAddressId)

Removes a Call Forwarding rule

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwReason

forwarding reason to remove Call Forwarding for (**RIL_FWDREASON_** constant)

dwInfoClasses

information classes to remove Call Forwarding for (combination of **RIL_INFOCLASS_** constants)

dwAddressId

address ID of line on which to change call forwarding settings

Comments

Asynchronous. *lpData* is *NULL*.

RIL_RemovePreferredOperator

HRESULT RIL_RemovePreferredOperator(HRIL hRil, DWORD dwIndex)

Removes a specified operator from the list of preferred operators

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwIndex

storage index of the preferred operator to remove

Comments

Asynchronous. *lpData* is *NULL*.

[illegible]

Parameters

 $hRil$

dwIndex

Comments

RIL SaveMsgConfig

Parameters

hRil

dwIndex

Comments

RIL SendDTMF

Parameters

hRil

lpszChars

dwDuration

Comments

RIL SendMsg

dwOptions)

107

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
	<i>lpMessage</i> message to be sent
	<i>dwOptions</i> options (any combination of <i>RIL_SENDOPT_</i> constants)
Comments	Asynchronous. <i>lpData</i> points to a DWORD containing the reference number of the sent message.

RIL_SendMsgAcknowledgement

HRESULT RIL_SendMsgAcknowledgement(HRIL hRil, BOOL fSuccess)

Sends an message acknowledgement

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
	<i>fSuccess</i> TRUE if success acknowledgment is to be sent; FALSE otherwise
Comments	Asynchronous. <i>lpData</i> is <i>NULL</i> . On Phase 2 mobiles, the radio automatically sends SMS message ACKs. But in Phase 2+, the MMI is responsible for these ACKs, hence this function.

RIL_SendRestrictedSimCmd

HRESULT RIL_SendRestrictedSimCmd(HRIL hRil, DWORD dwCommand, const RILSIMCMDPARAMETERS* lpParameters, const BYTE* lpbData, DWORD dwSize)

Sends a specified restricted command to the SIM

Parameters	<i>hRil</i> handle to RIL instance returned by RIL_Initialize
	<i>dwCommand</i> restricted command to be sent to the SIM (<i>RIL_SIMCMD_</i> constant)
	<i>lpParameters</i> Parameters for the command to be sent (can be <i>NULL</i> if parameters aren't required)
	<i>lpbData</i> Data to be written to the SIM (can be <i>NULL</i> if data isn't required)
	<i>dwSize</i> Size of the data pointed to by <i>lpbData</i> in bytes
Comments	Asynchronous. <i>lpData</i> points to an RILSIMRESPONSE structure.

RIL_SendSimCmd

HRESULT RIL_SendSimCmd(**HRIL** *hRil*, **const BYTE*** *lpbCommand*, **DWORD** *dwSize*)

Sends a specified command to the SIM

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpbCommand

command to be sent to the SIM

dwSize

size of the data pointed to by *lpbCommand* in bytes

Comments

Asynchronous. *lpData* points to an array of **BYTE**s.

RIL_SendSimToolkitCmdResponse

HRESULT RIL_SendSimToolkitCmdResponse(**HRIL** *hRil*, **const BYTE*** *lpbResponse*, **DWORD** *dwSize*)

Sends a response to an executed SIM Toolkit command

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpbResponse

response to be sent

dwSize

size of the data pointed to by *lpbResponse* in bytes

Comments

Asynchronous. *lpData* is **NULL**.

RIL_SendSimToolkitEnvelopeCmd

HRESULT RIL_SendSimToolkitEnvelopeCmd(**HRIL** *hRil*, **const BYTE*** *lpbCommand*, **DWORD** *dwSize*)

Sends a SIM Toolkit envelope command

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpbCommand

SIM Toolkit envelope command to be sent

dwSize

size of the data pointed to by *lpbCommand* in bytes

Comments

Asynchronous. *lpData* points to an array of **BYTE**s containing a response to the sent command.

RIL_SendStoredMsg

HRESULT RIL_SendStoredMsg(HRIL hRil, DWORD dwIndex, DWORD dwOptions)

Sends a message from the current storage location

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwIndex

index of the message to be sent

dwOptions

options (any combination of **RIL_SENDOPT_** constants)

Comments

Asynchronous. *lpData* points to a **DWORD** containing the reference number of the sent message. This feature is not used in Stinger and is untested.

RIL_SendSupServiceData

HRESULT RIL_SendSupServiceData(HRIL hRil, const BYTE* lpbData, DWORD dwSize)

Sends supplementary service (USSD) data

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpbData

data to be sent

dwSize

size of the data pointed to by *lpbData* in bytes

Comments

TBD

RIL_SetAudioDevices

HRESULT RIL_SetAudioDevices(HRIL hRil, const RILAUDIODEVICEINFO* lpAudioDeviceInfo)

Sets currently used transmit and receive audio devices

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpAudioDeviceInfo

audio devices to set

Comments

Asynchronous. *lpData* is **NULL**.

RIL_SetAudioGain

HRESULT RIL_SetAudioGain(HRIL hRil, const RILGAININFO* lpGainInfo)

Sets audio gain information

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpGainInfo

audio gain information to be sent

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetAudioMuting

HRESULT RIL_SetAudioMuting(HRIL hRil, BOOL fEnable)

Mutes or un-mutes the input audio device

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

fEnable

TRUE if input audio device is to be muted; FALSE otherwise

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetBearerServiceOptions

HRESULT RIL_SetBearerServiceOptions(HRIL hRil, const RILBEARERSVCINFO*

lpBearerServiceInfo)

Sets data bearer service options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpBearerServiceInfo

data bearer service options to set

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetCallBarringStatus

HRESULT RIL_SetCallBarringStatus(HRIL hRil, DWORD dwType, DWORD

dwInfoClass, LPCSTR *lpszPassword*, DWORD *dwStatus*)

Enables or disables the specified type of call barring

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwType
type of call barring to enable/disable (*RIL_BARRTYPE_* constant)

dwInfoClass
information class to enable/disable call barring for (*RIL_INFOCLASS_* constant)

lpzPassword
password to enable/disable call barring (can be *NULL* if password isn't required; no longer than *MAXLENGTH_PASSWORD* chars)

dwStatus
status to be set (*RIL_BARRINGSTATUS_* constant)

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetCallerIdStatus

HRESULT RIL_SetCallerIdStatus(HRIL hRil, DWORD dwStatus)

Sets the current CallerID status

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

dwStatus
status to be set (*RIL_SVCSTAT_* constant)

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetCallForwardingStatus

HRESULT RIL_SetCallForwardingStatus(HRIL hRil, DWORD dwReason, DWORD dwInfoClasses, DWORD dwStatus, DWORD dwAddressId)

Enables or disables the specified Call Forwarding rule

Parameters

hRil
handle to RIL instance returned by **RIL_Initialize**

dwReason
forwarding reason to enable/disable Call Forwarding for (*RIL_FWDREASON_* constant)

dwInfoClasses
information classes to enable/disable Call Forwarding for (combination of *RIL_INFOCLASS_* constants)

dwStatus
status to be set (*RIL_SVCSTAT_* constant)

dwAddressId
address ID of line on which to change call forwarding settings

Comments Asynchronous. *lpData* is *NULL*.

RIL SetCallWaitingStatus

HRESULT RIL_SetCallWaitingStatus(HRIL hRil, DWORD dwInfoClasses, DWORD dwStatus)

Enables or disables Call Waiting for the specified info class

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwInfoClasses

information classes to enable/disable Call Waiting for

dwStatus

status to be set (*RIL_SVCSTAT_* constant)

Comments

Asynchronous. *lpData* is *NULL*.

RIL SetClosedGroupSettings

HRESULT RIL_SetClosedGroupSettings(HRIL hRil, const RILCLOSEDGROUPSETTINGS* lpSettings)

Sets the Closed User Group settings

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpSettings

settings to be set

Comments

Asynchronous. *lpData* is *NULL*.

RIL SetCostInfo

HRESULT RIL_SetCostInfo(HRIL hRil, const RILCOSTINFO* lpCostInfo, LPCSTR lpszPassword)

Sets advice-of-charge settings

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpCostInfo

advice-of-charge settings to set

lpszPassword

password required to set advice-of-charge settings

Comments

Asynchronous. *lpData* points to a **RILCOSTINFO** structure. This feature is not used in Stinger and is untested.

RIL_SetCurrentAddressId

HRESULT RIL_SetCurrentAddressId(HRIL *hRil*, **DWORD** *dwAddressId*)

Sets the current address identifier (see RILSUBSCRIBERINFO)

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwAddressId

identifies the new addressID to use

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetDataCompression

HRESULT RIL_SetDataCompression(HRIL *hRil*, **const** RILDATACOMPINFO*

lpDataCompInfo)

Sets data compression options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpDataCompInfo

data compression options to set

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetDialedIdStatus

HRESULT RIL_SetDialedIdStatus(HRIL *hRil*, **DWORD** *dwStatus*)

Sets the current DialedID settings

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwStatus

status to be set (*RIL_SVCSTAT_* constant)

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetDTMFMonitoring

HRESULT RIL_SetDTMFMonitoring(HRIL *hRil*, **BOOL** *fEnable*)

Detects DTMF tones from an established voice call

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

fEnable

TRUE to initiate DTMF monitoring; FALSE to cancel

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetEquipmentState

HRESULT RIL_SetEquipmentState(**HRIL** *hRil*, **DWORD** *dwEquipmentState*)

Sets the equipment to the specified state

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwEquipmentState

equipment state to set (*RIL_EQSTATE_* constant)

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetErrorCorrection

HRESULT RIL_SetErrorCorrection(**HRIL** *hRil*, **const RILERRORCORRECTIONINFO*** *lpErrorCorrectionInfo*)

Set error correction options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpErrorCorrectionInfo

error correction options to set

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetGPRSAttached

HRESULT RIL_SetGPRSAttached(**HRIL** *hRil*, **BOOL** *fAttached*)

Sets the GPRS attach state

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

fAttached

TRUE: attached, FALSE: detached

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetGPRSClass

HRESULT RIL_SetGPRSClass(**HRIL** *hRil*, **DWORD** *dwClass*)

Sets the current GPRS class

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwClass
a **RIL_GPRSCLASS_*** constant

Comments Asynchronous. *lpData* is *NULL*

RIL_SetGPRSContext

HRESULT RIL_SetGPRSContext(**HRIL** *hRil*, **const RILGPRSCONTEXT*** *lpGprsContext*)

Sets a particular GPRS context

Parameters *hRil*
handle to RIL instance returned by **RIL_Initialize**

lpGprsContext
points to a **RILGPRSCONTEXT** structure

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetGPRSContextActivated

HRESULT RIL_SetGPRSContextActivated(**HRIL** *hRil*, **DWORD** *dwContextID*, **BOOL** *fContextActivation*)

Sets the GPRS activation state for a context

Parameters *hRil*
handle to RIL instance returned by **RIL_Initialize**

dwContextID
identifies the context

fContextActivation
TRUE: activated, FALSE: deactivated

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetHideConnectedIdStatus

HRESULT RIL_SetHideConnectedIdStatus(**HRIL** *hRil*, **DWORD** *dwStatus*)

Sets the current HideConnectedID settings

Parameters *hRil*
handle to RIL instance returned by **RIL_Initialize**

dwStatus
status to be set (**RIL_SVCSTAT_** constant)

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetHideIdStatus

HRESULT RIL_SetHideIdStatus(HRIL hRil, DWORD dwStatus)

Enables or disables HideID service

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwStatus

status to be set (*RIL_SVCSTAT_* constant)

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetHSCSDOptions

HRESULT RIL_SetHSCSDOptions(HRIL hRil, const RILHSCSDINFO* lpHscsdInfo)

Sets High Speed Circuit Switched Data options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpHscsdInfo

High Speed Circuit Switched Data options to set

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetLockingStatus

HRESULT RIL_SetLockingStatus(HRIL hRil, DWORD dwFacility, LPCSTR

lpszPassword, DWORD dwStatus)

Enables or disables locking status for the specified facility

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwFacility

facility to enable/disable locking for (*RIL_LOCKFACILITY_* constant)

lpszPassword

password to enable/disable locking (can be *NULL* if password isn't required; no longer than *MAXLENGTH_PASSWORD* chars)

dwStatus

status to be set (*RIL_LOCKINGSTATUS_* constant)

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetMinimumQualityOfService

HRESULT RIL_SetMinimumQualityOfService(HRIL hRil, const RILGPRSQOSPROFILE* lpGprsQosProfile)

Sets the minimum quality of service profile for a context

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpGprsQosProfile

points to a **RILGPRSQOSPROFILE** structure

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetMOSMSService

HRESULT RIL_SetMOSMSService(HRIL hRil, DWORD dwMoSmsService)

Sets the preferred SMS service option for mobile originated messages

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwMoSmsService

a **RIL_MOSMSSERVICE_*** constant

Comments

Asynchronous. *lpData* is *NULL*

RIL_SetMsgConfig

HRESULT RIL_SetMsgConfig(HRIL hRil, const RILMSGCONFIG*

lpMsgConfigInfo)

Sets messaging configuration

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpMsgConfigInfo

messaging configuration to be set

Comments

Asynchronous. *lpData* is *NULL*.

RIL_SetMsgServiceOptions

HRESULT RIL_SetMsgServiceOptions(HRIL hRil, const RILMSGSERVICEINFO*

lpMsgServiceInfo)

Sets messaging service options

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpMsgServiceInfo
messaging service options to be set

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetPhonebookOptions

HRESULT RIL_SetPhonebookOptions(**HRIL** *hRil*, **const RILPHONEBOOKINFO*** *lpPhonebookInfo*)
Sets phonebook options

Parameters *hRil*
handle to RIL instance returned by **RIL_Initialize**

lpPhonebookInfo
phonebook options to set

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetRequestedQualityOfService

HRESULT RIL_SetRequestedQualityOfService(**HRIL** *hRil*, **const RILGPRSQOSPROFILE*** *lpGprsQosProfile*)
Sets the requested quality of service profile for a context

Parameters *hRil*
handle to RIL instance returned by **RIL_Initialize**

lpGprsQosProfile
points to a **RILGPRSQOSPROFILE** structure

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetRLPOptions

HRESULT RIL_SetRLPOptions(**HRIL** *hRil*, **const RILRLPINFO*** *lpRlpInfo*)
Sets Radio Link Protocol options

Parameters *hRil*
handle to RIL instance returned by **RIL_Initialize**

lpRlpInfo
Radio Link Protocol options to set

Comments Asynchronous. *lpData* is *NULL*.

RIL_SetSimToolkitProfile

HRESULT RIL_SetSimToolkitProfile(**HRIL** *hRil*, **const BYTE*** *lpbProfile*, **DWORD** *dwSize*)

Sets SIM Toolkit terminal profile

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpbProfile

SIM Toolkit profile to be set

dwSize

size of the data pointed to by *lpbProfile* in bytes

Comments

Asynchronous. *lpData* is *NULL*.

RIL_TerminateSimToolkitSession

HRESULT RIL_TerminateSimToolkitSession(**HRIL** *hRil*, **DWORD** *dwCause*)

Terminates the SIM Toolkit session

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

dwCause

cause for session termination (*RIL_SIMTKITTERMCAUSE_* constant)

Comments

Asynchronous. *lpData* is *NULL*.

RIL_TransferCall

HRESULT RIL_TransferCall(**HRIL** *hRil*, **const RILADDRESS*** *lpAddress*, **const RILSUBADDRESS*** *lpSubAddress*)

Transfers incoming alerting call to the specified number

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpAddress

address to transfer the call to

lpSubAddress

sub-address to transfer the call to (can be *NULL*)

Comments

Asynchronous. *lpData* is *NULL*.

RIL UnlockPhone

HRESULT RIL_UnlockPhone(**HRIL** *hRil*, **LPCSTR** *lpzPassword*, **LPCSTR** *lpzNewPassword*)

Removes current lock applied to the phone

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpzPassword

password to unlock the phone (no longer than **MAXLENGTH_PASSWORD** chars)

lpzNewPassword

new password (can be **NULL**, unless the current locked state is one of the **RIL_LOCKEDSTATE_*_PUK** constants; no longer than **MAXLENGTH_PASSWORD** chars)

Comments

Asynchronous. *lpData* is **NULL**.

RIL UnregisterFromNetwork

HRESULT RIL_UnregisterFromNetwork(**HRIL** *hRil*)

Unregisters the ME from the current newtwork operator

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

Comments

Asynchronous. *lpData* is **NULL**.

RIL WriteMsg

HRESULT RIL_WriteMsg(**HRIL** *hRil*, **const RILMESSAGE*** *lpMessage*, **DWORD** *dwStatus*)

Writes a message to the current storage location

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpMessage

message to be written (of type **RIL_MSGTYPE_IN_DELIVER** or **RIL_MSGTYPE_OUT_SUBMIT**)

dwStatus

status to assigned to the written message (**RIL_MSGSTATUS_** constant)

Comments

Asynchronous. *lpData* points to a **DWORD** conaining the index used.

RIL_WritePhonebookEntry

HRESULT RIL_WritePhonebookEntry(HRIL hRil, const RILPHONEBOOKENTRY* lpEntry)

Writes a phonebook entry to the current storage location

Parameters

hRil

handle to RIL instance returned by **RIL_Initialize**

lpEntry

phonebook entry to write out

Comments

Asynchronous. *lpData* is *NULL*.

void (CALLBACK *RILNOTIFYCALLBACK)

typedef void (CALLBACK *RILNOTIFYCALLBACK)(DWORD dwCode, const void* lpData, DWORD cbData, DWORD dwParam)

RIL notification callback

Parameters

dwCode

notification code

lpData

data associated with the notification

cbData

size of the structure pointed to *lpData*

dwParam

parameter passed to **RIL_Initialize**

Comments

This function is called when the radio sends an unsolicited notification

void (CALLBACK *RILRESULTCALLBACK)

typedef void (CALLBACK *RILRESULTCALLBACK)(DWORD dwCode, HRESULT hrCmdID, const void* lpData, DWORD cbData, DWORD dwParam)

RIL function result callback

Parameters

dwCode

result code

hrCmdID

ID returned by the command that originated this response

lpData

data associated with the notification

cbData

size of the structure pointed to *lpData*

	<i>dwParam</i> parameter passed to RIL_Initialize
Comments	This function is called to send a return value after and asynchronous RIL function call

RILADDRESS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwType;
    DWORD dwNumPlan;
    WCHAR wszAddress[MAXLENGTH_ADDRESS];
} RILADDRESS;
```

	Represents a phone number
Members	cbSize structure size in bytes dwParams indicates valid parameters dwType type of number dwNumPlan numbering plan wszAddress[MAXLENGTH_ADDRESS] address (min 3, max 43)
Comments	None

RILAUDIODEVICEINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwTxDevice;
    DWORD dwRxDevice;
} RILAUDIODEVICEINFO;
```

	Audio device information
Members	cbSize structure size in bytes dwParams indicates valid parameters dwTxDevice transmit device

	dwRxDevice receive device
Comments	None

RILBEARERSVCINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwSpeed;
    DWORD dwServiceName;
    DWORD dwConnectionElement;
} RILBEARERSVCINFO;
```

Members Bearer service settings

cbSize
structure size in bytes

dwParams
indicates valid parameters

dwSpeed
offered data speed (protocol dependant)

dwServiceName
type of data service

dwConnectionElement
indicates transparent or non-transparent connection

Comments For *RIL_BSVCCCE_BOTH_* constants, the subsequent text indicates the preferred connection element. For instance, *RIL_BSVCCCE_BOTH_TRANSPARENT* means that both transparent and non transparent are supported, but transparent is preferred.

RILCALLERIDSETTINGS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwStatus;
    DWORD dwProvisioning;
} RILCALLERIDSETTINGS;
```

Members Caller ID settings

cbSize
structure size in bytes

dwParams
indicates valid parameters

	dwStatus activation status
	dwProvisioning network provisioning status
Comments	None

RILCALLFORWARDINGSETTING Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwStatus;
    DWORD dwInfoClasses;
    RILADDRESS raAddress;
    RILSUBADDRESS rsaSubAddress;
    DWORD dwDelayTime;
} RILCALLFORWARDINGSETTING;
```

Members

Call forwarding service settings

cbSize
structure size in bytes

dwParams
indicates valid parameters

dwStatus
activation status

dwInfoClasses
indicates which classes of calls to forward

raAddress
forwarding address

rsaSubAddress
forwarding subaddress

dwDelayTime
seconds to wait in *RIL_FWDREASON_NOREPLY* case

Comments None

RILCALLHSCSDINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwRxTimeslots;
    DWORD dwTxTimeslots;
    DWORD dwAirInterfaceUserRate;
    DWORD dwChannelCoding;
```


	<code>} RILCALLHSCSDINFO;</code>
	High speed circuit switched data information for the current call
Members	<p>cbSize structure size in bytes</p> <p>dwParams indicates valid parameters</p> <p>dwRxTimeslots number of receive timeslots currently in use</p> <p>dwTxTimeslots number of transmit timeslots currently in use</p> <p>dwAirInterfaceUserRate air interface user rate currently in use</p> <p>dwChannelCoding current channel coding</p>
Comments	None

RILCALLINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwID;
    DWORD dwDirection;
    DWORD dwStatus;
    DWORD dwType;
    DWORD dwMultiparty;
    RILADDRESS raAddress;
    WCHAR wszDescription[MAXLENGTH_DESCRIPTION];
} RILCALLINFO;
```

	Information about a specific call
Members	<p>cbSize structure size in bytes</p> <p>dwParams indicates valid parameters</p> <p>dwID identifies each call</p> <p>dwDirection incoming or outgoing</p> <p>dwStatus properties of the call</p> <p>dwType voice or data or fax</p>

	dwMultiparty conference call status
	raAddress call address
	wszDescription [MAXLENGTH_DESCRIPTION] any associated text
Comments	None

RILCALLWAITINGINFO Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwCallType; DWORD dwAddressId; RILREMOTEPARTYINFO rrpiCallerInfo; } RILCALLWAITINGINFO;</pre>
Members	<p>Call waiting info</p> <p>cbSize structure size in bytes</p> <p>dwParams indicates valid parameters</p> <p>dwCallType type of call</p> <p>dwAddressId indicates address ID on which the incoming call arrived (if available)</p> <p>rrpiCallerInfo caller information</p>
Comments	None

RILCAPSBARRINGPWDLENGTH Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwType; DWORD dwPasswordLength; } RILCAPSBARRINGPWDLENGTH;</pre>
Members	<p>Call barring password length capabilities</p> <p>cbSize structure size in bytes</p>

	dwParams indicates valid parameters
	dwType type of call barring
	dwPasswordLength maximum password length (TBD BUGBUG, should this be a range?)
Comments	None

RILCAPSBEARERSVC Structure

	typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwSpeeds1; DWORD dwSpeeds2; DWORD dwServiceNames; DWORD dwConnectionElements; } RILCAPSBEARERSVC;
	Bearer service capabilities
Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwSpeeds1 TBD
	dwSpeeds2 TBD
	dwServiceNames TBD
	dwConnectionElements TBD
Comments	None

RILCAPSDIAL Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwCallTypes;
    DWORD dwOptions;
} RILCAPSDIAL;
```

Dialing capabilities

Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwCallTypes type of call being placed
	dwOptions dialing options
Comments	None

RILCAPSHSCSD Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwMultislotClass;
    DWORD dwMaxRxTimeslots;
    DWORD dwMaxTxTimeslots;
    DWORD dwMaxTotalTimeslots;
    DWORD dwChannelCodings;
    DWORD dwAirInterfaceUserRates;
    RILRANGE rrTopRxTimeslotRange;
} RILCAPSHSCSD;
```

High Speed Circuit Switched Data capabilities	
Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwMultislotClass multislot class supported
	dwMaxRxTimeslots maximum number of receive timeslots
	dwMaxTxTimeslots maximum number of transmit timeslots
	dwMaxTotalTimeslots maximum number of total timeslots
	dwChannelCodings supported channel codings
	dwAirInterfaceUserRates supported air interfacerates
	rrTopRxTimeslotRange TBD
Comments	None

RILCAPSLOCKINGPWDLENGTH Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwFacility;
    DWORD dwPasswordLength;
} RILCAPSLOCKINGPWDLENGTH;
```

Members

Locking password length capabilities

cbSize

structure size in bytes

dwParams

indicates valid parameters

dwFacility

the locking facility

dwPasswordLength

maximum password length (TBD BUGBUG, should this be a range?)

Comments

None

RILCAPSMSGMEMORYLOCATIONS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwReadLocations;
    DWORD dwWriteLocations;
    DWORD dwStoreLocations;
} RILCAPSMSGMEMORYLOCATIONS;
```

Members

Message memory location capabilities

cbSize

structure size in bytes

dwParams

indicates valid parameters

dwReadLocations

supported read locations

dwWriteLocations

supported write locations

dwStoreLocations

supported store locations

Comments

None

RILCAPSRLP Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwVersion;
    RILRANGE rrIWSRange;
    RILRANGE rrMWSRange;
    RILRANGE rrAckTimerRange;
    RILRANGE rrRetransmissionAttsRange;
    RILRANGE rrReseqPeriodRange;
} RILCAPSRLP;
```

Members	Radio Link Protocol capabilities
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwVersion TBD
	rrIWSRange TBD
	rrMWSRange TBD
	rrAckTimerRange TBD
	rrRetransmissionAttsRange TBD
	rrReseqPeriodRange TBD
	Comments None

RILCELLTOWERINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwMobileCountryCode;
    DWORD dwMobileNetworkCode;
    DWORD dwLocationAreaCode;
    DWORD dwCellID;
    DWORD dwBaseStationID;
    DWORD dwBroadcastControlChannel;
    DWORD dwRxLevel;
    DWORD dwRxLevelFull;
```

```

        DWORD dwRxLevelSub;
        DWORD dwRxQuality;
        DWORD dwRxQualityFull;
        DWORD dwRxQualitySub;
        DWORD dwIdleTimeSlot;
        DWORD dwTimingAdvance;
        DWORD dwGPRSCellID;
        DWORD dwGPRSBaseStationID;
    } RILCELLTOWERINFO;

```

Members

Cell tower info

cbSize

structure size in bytes

dwParams

indicates valid parameters

dwMobileCountryCode

TBD

dwMobileNetworkCode

TBD

dwLocationAreaCode

TBD

dwCellID

TBD

dwBaseStationID

TBD

dwBroadcastControlChannel

TBD

dwRxLevel

Value from 0-63 (see GSM 05.08, 8.1.4)

dwRxLevelFull

Value from 0-63 (see GSM 05.08, 8.1.4)

dwRxLevelSub

Value from 0-63 (see GSM 05.08, 8.1.4)

dwRxQuality

Value from 0-7 (see GSM 05.08, 8.2.4)

dwRxQualityFull

Value from 0-7 (see GSM 05.08, 8.2.4)

dwRxQualitySub

Value from 0-7 (see GSM 05.08, 8.2.4)

dwIdleTimeSlot

TBD

dwTimingAdvance

TBD

dwGPRSCellID

TBD

	dwGPRSBaseStationID TBD
Comments	None

RILCLOSEDGROUPSETTINGS Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwStatus; DWORD dwIndex; DWORD dwInfo; } RILCLOSEDGROUPSETTINGS;</pre>
Members	<p>Close user group settings</p> <p>cbSize structure size in bytes</p> <p>dwParams indicates valid parameters</p> <p>dwStatus activation status</p> <p>dwIndex CUG index</p> <p>dwInfo additional CUG flags</p>
Comments	This feature is not used in Stinger and is untested.

RILCONNECTINFO Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwCallType; DWORD dwBaudRate; } RILCONNECTINFO;</pre>
Members	<p>Connection info</p> <p>cbSize structure size in bytes</p> <p>dwParams indicates valid parameters</p> <p>dwCallType type of the established connection (<i>RIL_CALLTYPE_</i> constant)</p>

	dwBaudRate Baud rate of the established connection (set only for <i>RIL_CALLTYPE_DATA</i>)
Comments	None

RILCOSTINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwCCM;
    DWORD dwACM;
    DWORD dwMaxACM;
    DWORD dwCostPerUnit;
    WCHAR wszCurrency[MAXLENGTH_CURRENCY];
} RILCOSTINFO;
```

Members

Service cost info

cbSize
structure size in bytes

dwParams
indicates valid parameters

dwCCM
current call meter

dwACM
accumulated call meter

dwMaxACM
maximum accumulated call meter

dwCostPerUnit
cost per unit, in 16.16 fixed point

wszCurrency[MAXLENGTH_CURRENCY]
current currency

Comments None

RILDATACOMPINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwDirection;
    DWORD dwNegotiation;
    DWORD dwMaxDictEntries;
    DWORD dwMaxStringLength;
} RILDATACOMPINFO;
```

Members	Data compression settings
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwDirection compression in transmit and/or receive directions
	dwNegotiation compression is required or optional
	dwMaxDictEntries maximum number of dictionary entries
	dwMaxStringLength maximum string length
Comments	None

RILDIALEDIDSETTINGS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwStatus;
    DWORD dwProvisioning;
} RILDIALEDIDSETTINGS;
```

Members	Dialed ID settings
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwStatus activation status
	dwProvisioning network provisioning status
Comments	None

RILDIALINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    HRESULT hrCmdId;
    DWORD dwCallId;
} RILDIALINFO;
```

Members	Ring information
	cbSize structure size in bytes
	dwParams indicates valid parameters
	hrCmdId handle of call being dialed
	dwCallId id of call being dialed
Comments	None

RILENTERGPRSDATAMODE Structure

Members	typedef struct { DWORD cbSize; } RILENTERGPRSDATAMODE;
	A quality of service profile
Comments	cbSize structure size in bytes
	None
Parameters	Disable "C4200: nonstandard extension used : zero-sized array in struct/union"
	<i>dwL2Protocol</i> an optional RILL2PROTOCOL_* constant
	<i>dwNumContexts</i> number of contexts which follow
	<i>dwContextID[]</i> identifies the context(s) to enter data state

RILEQUIPMENTINFO Structure

Members	typedef struct { DWORD cbSize; DWORD dwParams; char szManufacturer[MAXLENGTH_EQUIPINFO]; char szModel[MAXLENGTH_EQUIPINFO]; char szRevision[MAXLENGTH_EQUIPINFO]; char szSerialNumber[MAXLENGTH_EQUIPINFO]; } RILEQUIPMENTINFO;
	Equipment info
Members	cbSize structure size in bytes

	dwParams indicates valid parameters
	szManufacturer[MAXLENGTH_EQUIPINFO] manufacturer of the radio hardware
	szModel[MAXLENGTH_EQUIPINFO] model of the radio hardware
	szRevision[MAXLENGTH_EQUIPINFO] software version of the radio stack
	szSerialNumber[MAXLENGTH_EQUIPINFO] equipment identity (IMEI)
Comments	None

RILEQUIPMENTSTATE Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwRadioSupport; DWORD dwEqState; } RILEQUIPMENTSTATE;</pre>
Members	<p>Equipment state</p> <p>cbSize structure size in bytes</p> <p>dwParams indicates valid parameters</p> <p>dwRadioSupport RIL_RADIOSUPPORT_* Parameter</p> <p>dwEqState RIL_EQSTATE_* Parameter</p>
Comments	None

RILERRECTORRECTIONINFO Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwOriginalRequest; DWORD dwOriginalFallback; DWORD dwAnswererFallback; } RILERRECTORRECTIONINFO;</pre>
	Error correction settings

Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwOriginalRequest TBD
	dwOriginalFallback TBD
	dwAnswererFallback TBD
Comments	None

RILGAININFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwTxGain;
    DWORD dwRxGain;
} RILGAININFO;
```

Members	Audio gain information
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwTxGain transmit gain level (128 nominal, 0 automatic)
	dwRxGain receive gain level (128 nominal, 0 automatic)
Comments	None

RILGPRSANSWER Structure

```
typedef struct {
    DWORD cbSize;
} RILGPRSANSWER;
```

Members	A quality of service profile
	cbSize structure size in bytes
Comments	None Disable "C4200: nonstandard extension used : zero-sized array in struct/union"

Parameters

fAnswer
TRUE: accept, FALSE: reject

dwL2Protocol
an optional RIL2PROTOCOL_* constant

dwNumContexts
number of contexts which follow

dwContextID[]
identifies the context(s) to enter data state

RILGPRSCONTEXT Structure

```
typedef struct {  
    DWORD cbSize;  
    DWORD dwParams;  
    DWORD dwContextID;  
    DWORD dwProtocolType;  
    RILGPRSCONTEXT;  
    WCHAR wszAddress[MAXLENGTH_GPRSADDRESS];  
    DWORD dwDataCompression;  
    DWORD dwHeaderCompression;  
    DWORD dwParameterLength;  
    char szParameters[];  
} RILGPRSCONTEXT;
```

A PDP Context represents a certain configuration for packet data communication.

Members

cbSize
structure size in bytes

dwParams
indicates valid parameters

dwContextID
the context number

dwProtocolType
a RIL_GPRSPROTOCOL_*constant

RILGPRSCONTEXT
a logical name to select the gateway gprs (which defines the external packet data network to use)

wszAddress[MAXLENGTH_GPRSADDRESS]
the packet address to use (if null, request dynamic)

dwDataCompression
a RIL_GPRSDATACOMP_*

dwHeaderCompression
a RIL_GPRSHADERCOMP_*

dwParameterLength
length of parameters list

	szParameters[] parameters specific to the protocol type
Comments	None Disable "C4200: nonstandard extension used : zero-sized array in struct/union"

RILGPRSCONTEXTACTIVATED Structure

	typedef struct { DWORD cbSize; DWORD dwContextID; BOOL fActivated; } RILGPRSCONTEXTACTIVATED;
Members	Shows which contexts are active cbSize structure size in bytes dwContextID the context number fActivated whether the context is activated
Comments	None

RILGPRSPROTOCOLCAPS Structure

	typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwProtocolType; DWORD dwPrecedenceClass; DWORD dwDelayClass; DWORD dwReliabilityClass; DWORD dwPeakThruClass; DWORD dwMeanThruClass; } RILGPRSPROTOCOLCAPS;
Members	General Packet Radio Service capabilities cbSize structure size in bytes (padded to DWORD) dwParams indicates valid parameters dwProtocolType a RIL_GPRSPROTOCOL_* constant dwPrecedenceClass valid RIL_GPRSPRECEDENCECLASS_* constants

	dwDelayClass valid RIL_GPRSDELAYCLASS_* constants
	dwReliabilityClass valid RIL_GPRSRELIABILITYCLASS_* constants
	dwPeakThruClass valid RIL_GPRSPeakThruClass_* constants
	dwMeanThruClass valid RIL_GPRSMeanThruClass_* constants
Comments	TBDTBD Disable "C4200: nonstandard extension used : zero-sized array in struct/union"

RILGPRSPROTOCOLCAPS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwProtocolType;
    RILRANGE ContextIDRange;
    DWORD dwDataCompression;
    DWORD dwHeaderCompression;
    DWORD dwParameterLength;
    char szParameters[];
} RILGPRSPROTOCOLCAPS;
```

Members

General Packet Radio Service capabilities

cbSize
structure size in bytes (padded to DWORD)

dwParams
indicates valid parameters

dwProtocolType
a RIL_GPRSPROTOCOL_* constant

ContextIDRange
min/max context ids

dwDataCompression
valid data compression values

dwHeaderCompression
valid header compression values

dwParameterLength
length of parameters list in bytes

szParameters[]
valid string parameters of this protocol type, delimited by , with final param terminated by

Comments	TBDTBD Disable "C4200: nonstandard extension used : zero-sized array in struct/union"
-----------------	--

RILGPRSQOSPROFILE Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwContextID;
    DWORD dwPrecedenceClass;
    DWORD dwDelayClass;
    DWORD dwReliabilityClass;
    DWORD dwPeakThruClass;
    DWORD dwMeanThruClass;
} RILGPRSQOSPROFILE;
```

Members	A quality of service profile
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwContextID the context number
	dwPrecedenceClass a RIL_GPRSPRECEDENCECLASS_* constant
	dwDelayClass a RIL_GPRSDELAYCLASS_* constant
	dwReliabilityClass a RIL_GPRSRELIABILITYCLASS_* constant
	dwPeakThruClass a RIL_GPRSPPEAKTHRUCLASS_* constant
Comments	dwMeanThruClass a RIL_GPRSMEANTHRUCLASS_* constant
	None

RILHIDECONNECTEDIDSETTINGS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwStatus;
    DWORD dwProvisioning;
} RILHIDECONNECTEDIDSETTINGS;
```

Members	Hide Connected ID settings
	cbSize structure size in bytes

	dwParams indicates valid parameters
	dwStatus activation status
	dwProvisioning network provisioning status
Comments	None

RILHIDEIDSETTINGS Structure

	typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwStatus; DWORD dwProvisioning; } RILHIDEIDSETTINGS;
	Hide ID settings
Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwStatus activation status
	dwProvisioning network provisioning status
Comments	None

RILHSCSDINFO Structure

	typedef struct { DWORD cbSize; DWORD dwParams; DWORD dwTranspRxTimeslots; DWORD dwTranspChannelCodings; DWORD dwNonTranspRxTimeslots; DWORD dwNonTranspChannelCodings; DWORD dwAirInterfaceUserRate; DWORD dwRxTimeslotsLimit; BOOL fAutoSvcLevelUpgrading; } RILHSCSDINFO;
	High speed circuit switched data settings

Members

cbSize	structure size in bytes
dwParams	indicates valid parameters
dwTranspRxTimeslots	number of receive timeslots for transparent HSCSD calls
dwTranspChannelCodings	accepted channel codings for transparent HSCSD calls
dwNonTranspRxTimeslots	number of receive timeslots for non-transparent HSCSD calls
dwNonTranspChannelCodings	accepted channel codings for non-transparent HSCSD calls
dwAirInterfaceUserRate	air interface user rate for non-transparent HSCSD calls
dwRxTimeslotsLimit	maximum number of receive timeslots to be used during the next non-transparent HSCSD call
fAutoSvcLevelUpgrading	TRUE if automatic user-initiated service level upgrading for non-transparent HSCSD calls is enabled, FALSE otherwise

Comments

None

RILMESSAGE Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    RILADDRESS raSvcCtrAddress;
    DWORD dwType;
    DWORD dwFlags;
    union {
        struct {
            RILADDRESS raOrigAddress;
            DWORD dwProtocolID;
            RILMSGDCS rmdDataCoding;
            SYSTEMTIME stSCReceiveTime;
            DWORD cbHdrLength;
            DWORD cchMsgLength;
            BYTE rgbHdr[MAXLENGTH_HDR];
            BYTE rgbMsg[MAXLENGTH_MSG];
            msgInDeliver;
        };
        struct {
            DWORD dwTgtMsgReference;
            RILADDRESS raTgtRecipAddress;
            SYSTEMTIME stTgtSCReceiveTime;
            SYSTEMTIME stTgtDischargeTime;
        };
    };
};
```

```

        DWORD dwTgtDlvStatus;
        DWORD dwProtocolID;
        RILMSGDCS rmdDataCoding;
        DWORD cbHdrLength;
        DWORD cchMsgLength;
        BYTE rgbHdr[MAXLENGTH_HDR];
        BYTE rgbMsg[MAXLENGTH_MSG];
        msgInStatus;
        struct {
            RILADDRESS raDestAddress;
            DWORD dwProtocolID;
            RILMSGDCS rmdDataCoding;
            DWORD dwVPFormat;
            SYSTEMTIME stVP;
            DWORD cbHdrLength;
            DWORD cchMsgLength;
            BYTE rgbHdr[MAXLENGTH_HDR];
            BYTE rgbMsg[MAXLENGTH_MSG];
            msgOutSubmit;
            struct {
                DWORD dwProtocolID;
                DWORD dwCommandType;
                DWORD dwTgtMsgReference;
                RILADDRESS raDestAddress;
                DWORD cbCmdLength;
                BYTE rgbCmd[MAXLENGTH_CMD];
                msgOutCommand;
                struct {
                    DWORD dwGeoScope;
                    DWORD dwMsgCode;
                    DWORD dwUpdateNumber;
                    DWORD dwID;
                    RILMSGDCS rmdDataCoding;
                    DWORD dwTotalPages;
                    DWORD dwPageNumber;
                    DWORD cchMsgLength;
                    BYTE rgbMsg[MAXLENGTH_MSG];
                    msgBcGeneral;
                    DWORD cchMsgLength;
                    BYTE rgbMsg[MAXLENGTH_MSG];
                    msgOutRaw;
                }
            }
        } RILMESSAGE;

```

Members

Message data

cbSize

structure size in bytes

dwParams

indicates valid parameters

raSvcCtrAddress

service center address

```

dwType
    type of message
dwFlags
    message flags

    UNION MEMBER

        RIL_MSGTYPE_IN_DELIVER
raOrigAddress
    originating address
dwProtocolID
    message protocol
rmdDataCoding
    data coding scheme
stSCReceiveTime
    receive time
cbHdrLength
    length of header in bytes
cchMsgLength
    length of body in bytes
rgbHdr[MAXLENGTH_HDR]
    header buffer
rgbMsg[MAXLENGTH_MSG]
    body buffer
msgInDeliver
    End RIL_MSGTYPE_IN_DELIVER

        RIL_MSGTYPE_IN_STATUS
dwTgtMsgReference
    target message reference
raTgtRecipAddress
    receipient address
stTgtSCReceiveTime
    receipient receive time
stTgtDischargeTime
    receipient dischage time
dwTgtDlvStatus
    delivery status
dwProtocolID
    message protocol
rmdDataCoding
    data coding scheme

```

cbHdrLength
length of header in bytes

cchMsgLength
length of body in bytes

rgbHdr[MAXLENGTH_HDR]
header buffer

rgbMsg[MAXLENGTH_MSG]
body buffer

msgInStatus
End RIL_MSGTYPE_IN_STATUS

RIL_MSGTYPE_OUT_SUBMIT

raDestAddress
destination address

dwProtocolID
message protocol

rmdDataCoding
data coding scheme

dwVPFormat
TBD

stVP
TBD

cbHdrLength
length of header in bytes

cchMsgLength
length of body in bytes

rgbHdr[MAXLENGTH_HDR]
header buffer

rgbMsg[MAXLENGTH_MSG]
body buffer

msgOutSubmit
RIL_MSGTYPE_OUT_SUBMIT

RIL_MSGTYPE_OUT_COMMAND

dwProtocolID
message protocol

dwCommandType
command type

dwTgtMsgReference
target message reference

raDestAddress
destination address

cbCmdLength
length of command in bytes

rgbCmd[MAXLENGTH_CMD]
command buffer

msgOutCommand
RIL_MSGTYPE_OUT_COMMAND

RIL_MSGTYPE_BC_GENERAL

dwGeoScope
message protocol

dwMsgCode
message code

dwUpdateNumber
update number

dwID
identity

rmdDataCoding
data coding scheme

dwTotalPages
total number of pages

dwPageNumber
current page number

cchMsgLength
length of message in bytes

rgbMsg[MAXLENGTH_MSG]
message buffer

msgBcGeneral
RIL_MSGTYPE_BC_GENERAL

cchMsgLength
length of body in bytes

rgbMsg[MAXLENGTH_MSG]
message buffer

msgOutRaw
RIL_MSGTYPE_OUT_RAW

Comments

None

RILMESSAGE IN SIM Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwLocation;
    DWORD dwIndex;
```

```
} RILMESSAGE_IN_SIM;
```

Members

Message data in sim info

cbSize

structure size in bytes

dwLocation

storage area (one of RIL_MSGLOC_XXXX)

dwIndex

storage index occupied by the message

Comments

None

RILMESSAGEINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwIndex;
    DWORD dwStatus;
    RILMESSAGE rmMessage;
} RILMESSAGEINFO;
```

Members

Message data with additional info

cbSize

structure size in bytes

dwParams

indicates valid parameters

dwIndex

storage index occupied by the message

dwStatus

message status

rmMessage

the message itself

Comments

None

RILMSGCONFIG Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    RILADDRESS raSvcCtrAddress;
    char szBroadcastMsgIDs[MAXLENGTH_MSGIDS];
    DWORD dwBroadcastMsgLangs;
} RILMSGCONFIG;
```


Members	Messaging configuration
	cbSize structure size in bytes
	dwParams indicates valid parameters
	raSvcCtrAddress service center address
	szBroadcastMsgIDs[MAXLENGTH_MSGIDS] list of subscribed broadcast message IDs
	dwBroadcastMsgLangs broadcast message languages
Comments	None

RILMSGDCS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwType;
    DWORD dwFlags;
    DWORD dwMsgClass;
    DWORD dwAlphabet;
    DWORD dwIndication;
    DWORD dwLanguage;
} RILMSGDCS;
```

Members	Message data coding scheme
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwType DCS type
	dwFlags DCS flags
	dwMsgClass message class (Only for RIL_DCSTYPE_GENERAL and RIL_DCSTYPE_MSGCLASS)
	dwAlphabet DCS alphabet
	dwIndication indication (Only for RIL_DCSTYPE_MSGWAIT)
	dwLanguage indication (Only for RIL_DCSTYPE_LANGUAGE)

RILMSGSERVICEINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwService;
    DWORD dwMsgClasses;
    DWORD dwReadLocation;
    DWORD dwReadUsed;
    DWORD dwReadTotal;
    DWORD dwWriteLocation;
    DWORD dwWriteUsed;
    DWORD dwWriteTotal;
    DWORD dwStoreLocation;
    DWORD dwStoreUsed;
    DWORD dwStoreTotal;
} RILMSGSERVICEINFO;
```

Members

Messaging service settings

cbSize

structure size in bytes

dwParams

indicates valid parameters

dwService

supported service types

dwMsgClasses

supported message classes

dwReadLocation

current read location

dwReadUsed

number of fields used

dwReadTotal

total number of fields

dwWriteLocation

current read location

dwWriteUsed

number of fields used

dwWriteTotal

total number of fields

dwStoreLocation

current read location

dwStoreUsed

number of fields used

	dwStoreTotal total number of fields
Comments	None

RILMSGSTORAGEINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwReadLocation;
    DWORD dwWriteLocation;
    DWORD dwStoreLocation;
} RILMSGSTORAGEINFO;
```

	Message storage locations
Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwReadLocation current read location
	dwWriteLocation current write location
	dwStoreLocation current store location

Comments	None
-----------------	------

RILOPERATORINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwIndex;
    DWORD dwStatus;
    RILOPERATORNAMES ronNames;
} RILOPERATORINFO;
```

	Indicates status of a particular operator
Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwIndex index, if applicable

	dwStatus	registration status, if applicable
	ronNames	representations of an operator
Comments		None

RILOPERATORNAMES Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    char szLongName[MAXLENGTH_OPERATOR_LONG];
    char szShortName[MAXLENGTH_OPERATOR_SHORT];
    char szNumName[MAXLENGTH_OPERATOR_NUMERIC];
} RILOPERATORNAMES;
```

	The different representations of an operator	
Members	cbSize	structure size in bytes
	dwParams	indicates valid parameters
	szLongName[MAXLENGTH_OPERATOR_LONG]	long representation (max 16 characters)
	szShortName[MAXLENGTH_OPERATOR_SHORT]	short representation (max 8 characters)
	szNumName[MAXLENGTH_OPERATOR_NUMERIC]	numeric representation (3 digit country code & 2 digit network code)
Comments		None

RILPHONEBOOKENTRY Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwIndex;
    RILADDRESS raAddress;
    WCHAR wszText[MAXLENGTH_PHONEBOOKTEXT];
} RILPHONEBOOKENTRY;
```

	A single phonebook entry	
Members	cbSize	structure size in bytes
	dwParams	indicates valid parameters

dwIndex
index of the entry

raAddress
the stored address

wszText[MAXLENGTH_PHONEBOOKTEXT]
associated text

Comments None

RILPHONEBOOKINFO Structure

```
typedef struct {  
    DWORD cbSize;  
    DWORD dwParams;  
    DWORD dwStoreLocation;  
    DWORD dwUsed;  
    DWORD dwTotal;  
} RILPHONEBOOKINFO;
```

Phonebook settings

cbSize
structure size in bytes

dwParams
indicates valid parameters

dwStoreLocation
location of phonebook memory

dwUsed
number of locations used

dwTotal
total number of phonebook locations

Members

Comments None

RILRANGE Structure

```
typedef struct {  
    DWORD dwMinValue;  
    DWORD dwMaxValue;  
} RILRANGE;
```

Range of values

dwMinValue
minimum value

dwMaxValue
maximum value

Members

Comments None

RILREMOTEPARTYINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    RILADDRESS raAddress;
    RILSUBADDRESS rsaSubAddress;
    WCHAR wszDescription[MAXLENGTH_DESCRIPTION];
    DWORD dwValidity;
} RILREMOTEPARTYINFO;
```

Members

Incoming call info

cbSize
structure size in bytes

dwParams
indicates valid parameters

raAddress
address of caller

rsaSubAddress
subaddress of caller

wszDescription[MAXLENGTH_DESCRIPTION]
text associated with caller

dwValidity
indicates validity of caller info

Comments None

RILRINGINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwCallType;
    DWORD dwAddressId;
    RILSERVICEINFO rsiServiceInfo;
} RILRINGINFO;
```

Members

Ring information

cbSize
structure size in bytes

dwParams
indicates valid parameters

	dwCallType type of the offered call (<i>RIL_CALLTYPE_</i> constant)
	dwAddressId indicates address ID on which the incoming call arrived (if available)
	rsiServiceInfo data connection service information (set only for <i>RIL_CALLTYPE_DATA</i>)
Comments	None

RILRLPINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwIWS;
    DWORD dwMWS;
    DWORD dwAckTimer;
    DWORD dwRetransmissionAttempts;
    DWORD dwVersion;
    DWORD dwResequencingPeriod;
} RILRLPINFO;
```

	Radio link protocol settings
Members	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwIWS IWF-to-MS window size
	dwMWS MS-to-IWF window size
	dwAckTimer acknowledgement timer in 10s of milliseconds (T1)
	dwRetransmissionAttempts number of retransmission attempts (N2)
	dwVersion RLP version number
	dwResequencingPeriod resequencing period (T4)
Comments	None

RILSERIALPORTSTATS Structure

```
typedef struct {
```

```

        DWORD cbSize;
        DWORD dwParams;
        DWORD dwReadBitsPerSecond;
        DWORD dwWrittenBitsPerSecond;
    } RILSERIALPORTSTATS;

```

Members	Statistics of the virtual serial port
	cbSize structure size in bytes
	dwParams indicates valid parameters
	dwReadBitsPerSecond bit rate for reading data
	dwWrittenBitsPerSecond bit rate for writing data
Comments	None

RILSERVICEINFO Structure

```

typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    BOOL fSynchronous;
    BOOL fTransparent;
} RILSERVICEINFO;

```

Members	Connection service information
	cbSize structure size in bytes
	dwParams indicates valid parameters
	fSynchronous TRUE if connection service is synchronous, FALSE if asynchronous
	fTransparent TRUE if connection service is transparent, FALSE if non-transparent
Comments	None

RILSIGNALQUALITY Structure

```

typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    int nSignalStrength;
    int nMinSignalStrength;
}

```



```

        int nMaxSignalStrength;
        DWORD dwBitErrorRate;
        int nLowSignalStrength;
        int nHighSignalStrength;
    } RILSIGNALQUALITY;

```

Members

Signal quality info

cbSize
structure size in bytes

dwParams
indicates valid parameters

nSignalStrength
TBD

nMinSignalStrength
TBD

nMaxSignalStrength
TBD

dwBitErrorRate
bit error rate in 1/100 of a percent

nLowSignalStrength
TBD

nHighSignalStrength
TBD

Comments

None

RILSIMCMDPARAMETERS Structure

```

typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwFileID;
    DWORD dwParameter1;
    DWORD dwParameter2;
    DWORD dwParameter3;
} RILSIMCMDPARAMETERS;

```

Members

Parameters for a restricted SIM command

cbSize
structure size in bytes

dwParams
indicates valid parameters

dwFileID
SIM file ID

dwParameter1
parameter specific to SIM command

	dwParameter2
	parameter specific to SIM command
	dwParameter3
	parameter specific to SIM command
Comments	None

RILSIMRECORDSTATUS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwRecordType;
    DWORD dwItemCount;
    DWORD dwSize;
} RILSIMRECORDSTATUS;
```

	Response to a restrcited SIM command
Members	cbSize
	Size of the structure in bytes
	dwParams
	Indicates valid parameter values
	dwRecordType
	RIL_SIMRECORDTYPE_* Constant
	dwItemCount
	Number of items in the record
	dwSize
	Size in bytes of each item
Comments	None

RILSIMRESPONSE Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwStatusWord1;
    DWORD dwStatusWord2;
    BYTE pbResponse[];
} RILSIMRESPONSE;
```

	Response to a restrcited SIM command
Members	cbSize
	structure size in bytes
	dwParams
	indicates valid parameters

	dwStatusWord1	return parameter specific to SIM command
	dwStatusWord2	return parameter specific to SIM command
	pbResponse[]	additional bytes of response data
Comments	None	Disable "C4200: nonstandard extension used : zero-sized array in struct/union"

RILSIMTOOLKITNOTIFYCAPS Structure

	<pre>typedef struct { DWORD cbSize; DWORD dwParams; } RILSIMTOOLKITNOTIFYCAPS;</pre>	
	SIM TOOLKIT NOTIFY CAPABILITIES	
Members	cbSize	structure size in bytes
	dwParams	indicates valid parameters
Comments	This structure indicates who implements the various SIM ToolKit Notifications	
Values	dwRefresh	<i>TBD</i>
	dwMoreTime	<i>TBD</i>
	dwPollInterval	<i>TBD</i>
	dwPollingOff	<i>TBD</i>
	dwSetUpCall	<i>TBD</i>
	dwSendSS	<i>TBD</i>
	dwSendSMS	<i>TBD</i>
	dwPlayTone	<i>TBD</i>
	dwDisplayText	<i>TBD</i>
	dwGetInkey	<i>TBD</i>

TBD

TBD

TBD

TBD

RILSUBADDRESS Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwType;
    WCHAR wszSubAddress[MAXLENGTH_SUBADDR];
} RILSUBADDRESS;
```

The subaddress of a called party

cbSize

structure size in bytes

dwParams

indicates valid parameters

dwType

type of subaddress

wszSubAddress[MAXLENGTH_SUBADDR]

subaddress (min 2, max 23)

Comments

None

RILSUBSCRIBERINFO Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    RILADDRESS raAddress;
    WCHAR wszDescription[MAXLENGTH_DESCRIPTION];
    DWORD dwSpeed;
    DWORD dwService;
    DWORD dwITC;
    DWORD dwAddressId;
} RILSUBSCRIBERINFO;
```

A phone number assigned to the user

Members	cbSize
	structure size in bytes
	dwParams
	indicates valid parameters
	raAddress
	the assigned address
	wszDescription [MAXLENGTH_DESCRIPTION]
	text relating to this subscriber
	dwSpeed
	data rate related to this number
	dwService
	the service related to this number
	dwITC
	information transfer capability
	dwAddressId
	the address ID of this number

Comments None

RILSUPSERVICEDATA Structure

```
typedef struct {
    DWORD cbSize;
    DWORD dwParams;
    DWORD dwStatus;
    BYTE pbData[];
} RILSUPSERVICEDATA;
```

Members	Supplementary service data
	cbSize
	structure size in bytes
	dwParams
	indicates valid parameters
	dwStatus
	additional status for message
	pbData []
	message itself

Comments None
 Disable "C4200: nonstandard extension used : zero-sized array in struct/union"